Minimal proper non-IRUP instances of the one-dimensional Cutting Stock Problem

Vadim M. Kartak[1], Artem V. Ripatti[2], Guntram Scheithauer[3] and Sascha Kurz[4]

[1]Bashkir State Pedagogical University, Russia
[2]Ufa State Aviation Technical University, Russia
[3]Technische Universität Dresden, Germany
[4]Universität Bayreuth, Germany

November 12, 2017

Abstract: We consider the well-known one dimensional cutting stock problem (1CSP). Based on the pattern structure of the classical ILP formulation of Gilmore and Gomory, we can decompose the infinite set of 1CSP instances, with a fixed demand n, into a finite number of equivalence classes. We show up a strong relation to weighted simple games. Studying the integer round-up property we computationally show that all 1CSP instances with $n \leq 9$ are proper IRUP, while we give examples of a proper non-IRUP instances with $n = 10$. A gap larger than 1 occurs for $n = 11$. The worst known gap is raised from 1.003 to 1.0625. The used algorithmic approaches are based on exhaustive enumeration and integer linear programming. Additionally we give some theoretical bounds showing that all 1CSP instances with some specific parameters have the proper IRUP.

Keywords: bin packing problem, cutting stock problem, integer round-up property, equivalence of instances, branch and bound method, linear programming, weighted simple games.

1 Introduction

Baum and Trotter [1] have introduced the notation of having the integer round-up property (IRUP) for integer linear minimization problems (ILPs), stating that rounding up the optimal value of its LP relaxation yields an upper bound. Here we study the one-dimensional cutting stock problem (1CSP) with respect to the IRUP. The classical ILP formulation for the cutting stock problem by Gilmore and Gomory is based on so-called cutting patterns [8]. Using this formulation, Marcotte [15] has shown that certain subclasses of cutting stock problems have the IRUP, while she later showed that there are instances of 1CSP having a gap of exactly 1 [16]. The first example with gap larger than 1 was given in [6]. Subsequently, the gap was increased to $\frac{6}{5}$ [18, 21, 24], i.e., no example with a gap of at least 2 is currently known. Indeed, the authors of [23] have conjectured that the gap is always below 2 – called the MIRUP property, which is one of the most important theoretical issues about 1CSP, see also [5]. Practical experience shows that the typical gap is rather small [22]. An algorithm for verifying when an 1CSP instance does not have IRUP is presented in [10].

Dropping some cutting patterns from the ILP formulation of [8], the authors of [17] introduced the *proper relaxation*, which is at most as worse as the LP relaxation. We call the difference of the optimal value and the optimal value of the corresponding proper relaxation the proper gap. If the proper gap is at least 1, we speak of a proper non-IRUP instance. The currently largest known proper gap of 1CSP is given by 1.003 [2, 4].

The paper is organized as follows. In Section 2 we introduce the basic notation. The concept of partitioning the infinite set of 1CSP with fixed demand into a finite set of equivalence classes is described in Section 3. The relation to the discrete structure of weighted simple games, from cooperative game theory, is the topic of Section 4. In Section 5 we develop an exhaustive enumeration algorithm for the generation of all equivalence classes of 1CSP instances. We proceed with some theoretical bounds on the proper gap of 1CSP instances in Section 6. Based on the integer linear programming approaches in Section 7 we present computational results in Section 8. We draw a conclusion in Section 9.

2 Basic notation

Assume that one-dimensional material objects like, e.g., paper reels or wooden rods of a given length $L \in \mathbb{R}_{>0}$ are cut into smaller pieces of lengths $l_1, \dots, l_m \in \mathbb{R}_{>0}$ in order to fulfill the order demands $b_1, \dots, b_m \in \mathbb{Z}_{>0}$. The question for the minimum needed total amount of stock material or, equivalently, the minimization of waste, is the famous 1CSP. Using the abbreviations $l = (l_1, \dots, l_m)^T$ and $b = (b_1, \dots, b_m)^T$ we denote an instance of 1CSP by $E = (m, L, l, b)$. W.l.o.g. we assume $0 < l_1 \leq \dots \leq l_m \leq L$ in the following.

The cutting patterns, mentioned in the introduction, are formalized as vectors $a = (a_1, \dots, a_m)^\top \in \mathbb{Z}_{\geq 0}^m$. We call a *pattern* (of E) *feasible* if $l^\top a \leq L$. By $P^f(E) := \left\{a \,:\, l^\top a \leq L,\, a \in \mathbb{Z}_{\geq 0}^m\right\}$ we denote the set of all feasible patterns. Additionally, we call a pattern *proper*, if it is feasible and $a_i \leq b_i$ for all $1 \leq i \leq m$. By $P^p(E) := \left\{a \,:\, l^\top a \leq L,\, a \in \mathbb{Z}_{\geq 0}^m,\, a_i \leq b_i,\, 1 \leq i \leq m\right\}$ we denote the set of all proper patterns.

Given a set of patterns $P = \{a^1, \dots, a^r\}$ (of E), let $A(P)$ denote the concatenation of the pattern vectors a^i. With this we can define

$$z_D(P, E) := \sum_{i=1}^{r} x_i \to \min \quad \text{subject to} \quad A(P)x = b,\ x \in \mathbb{Z}_{\geq 0}^r \quad \text{and}$$

$$z_C(P, E) := \sum_{i=1}^{r} x_i \to \min \quad \text{subject to} \quad A(P)x = b,\ x \in \mathbb{R}_{\geq 0}^r.$$

Choosing $P = P^f(E)$ we obtain the mentioned ILP formulation for 1CSP of Gilmore and Gomory and its continuous relaxation. As abbreviations we use $z_D^f(E) := z_D(P^f(E), E)$, $z_C^f(E) := z_C(P^f(E), E)$, and $\Delta(E) := z_D^f(E) - z_C^f(E)$, where the later is called the *gap of instance* E. So an instance E has *IRUP* if $\Delta(E) < 1$ and is called *non-IRUP instance* otherwise. *MIRUP instances* are those with $\Delta(E) < 2$.

Choosing $P = P^p(E)$ we obtain the proper relaxation with optimal value $z_C^p(E) := z_C(P^p(E), E)$. Since $z_D(P^p(E), E) = z_D(P^f(E), E)$, we call $\Delta_p(E) := z_D^f(E) - z_C^p(E)$ the *proper gap of instance* E. Similarly, an instance E is a *proper IRUP instance* if $\Delta_p(E) < 1$ and a *proper non-IRUP instance* otherwise.

Due to $\Delta_p(E) \leq \Delta(E)$ proper non-IRUP instances are also non-IRUP instances. The converse is not true as shown by $E = \left(3, 30, (2,3,5)^\top, (1,2,4)^\top\right)$ with $\Delta(E) = 31/30$ and $\Delta_p(E) = 4/5$.

3 Equivalence of 1CSP instances

Given an 1CSP instance $E = (m, L, l, b)$ with a demand of $n = \sum_{i=1}^{m} b_i$, we can easily transform it into an instance $\bar{E} = (n, L, l', b')$ with $b'_i = 1$ for all $1 \leq i \leq n$ by taking b_j copies of length l_j for each $1 \leq j \leq m$. We can easily check that this has no effect on the stated (I)LPs, i.e., we have $z_D^f(E) = z_D^f(\bar{E})$, $z_C^f(E) = z_C^f(\bar{E})$, and $z_C^p(E) = z_C^p(\bar{E})$. Thus we assume $b_i = 1$ and abbreviate 1CSP instances by $E = (n, L, l)$ in the following. Especially we have $P^p(E) \in \mathbb{B}^n$, where $\mathbb{B} = \{0, 1\}$.

We remark that, using this modification, the 1CSP becomes equivalent to the Bin Packing Problem (BPP), where n items of size l_i have to be packed into as few as possible identical bins of capacity L. So our results also hold for the BPP and indeed some part of the related literature considers the BPP instead of the 1CSP. The continuous relaxation of BPP is also known as the Fractional Bin Packing Problem, cf. [2].

Since the set partitioning formulation of Gilmore and Gomory and its proper relaxation actually do not depend directly on the parameters L and l_i, we partition the set of 1CSP instances for a fixed demand n according to their set of proper patterns.

Definition 1. *1CSP instances E and $\bar{E}$ are pattern-equivalent if $P^p(E) = P^p(\bar{E})$.*

Since $z_C^p(E) = z_C^p(\bar{E})$, $z_D^f(E) = z_D^f(\bar{E})$, and $\Delta_p(E) = \Delta_p(\bar{E})$ for pattern-equivalent instances E and $\bar{E}$, we can restrict ourselves onto the set of equivalence classes

$$\mathbb{P}_n^p := \left\{ P^p(E) \,:\, E = (n, L, l),\, L \in \mathbb{R}_{>0},\, l \in \mathbb{R}_{>0}^n \right\}.^1$$

Obviously, $|\mathbb{P}_n^p| \leq 2^{|\mathbb{B}^n|} = 2^{2^n}$ is finite, but not all subsets of $\mathbb{B}^n$ can be attained as proper patterns of an 1CSP instance.

Lemma 1. *Given $P \subseteq \mathbb{B}^n$, an 1CSP instance $E = (n, L, l)$ with $P^p(E) = P$ exists iff the following system of linear inequalities contains a solution:*

$$\begin{aligned} 1 \leq l_1 \leq \cdots \leq l_n \leq L, & \qquad\qquad (1) \\ \sum_{i=1}^{n} l_i a_i \leq L \qquad & \forall a \in P, \\ \sum_{i=1}^{n} l_i a_i \geq L + 1 \qquad & \forall a \in \mathbb{B}^n \setminus P, \\ l_1, l_2, \ldots l_n, L \in \mathbb{R}_{\geq 0}. & \end{aligned}$$

[1] We remark that the authors of [19] have introduced a finer equivalence relation, called full pattern-equivalence and based on $P^f(E) = P^f(\bar{E})$, which is needed if also $z_C^f(E)$ and $\Delta(E)$ should be preserved. For example, the instances $E = (6, 30, (6, 6, 10, 10, 11, 15)^\top)$ and $\bar{E} = (6, 10000, (2000, 2000, 3001, 3250, 3750, 5000)^\top)$ are proper pattern-equivalent but not full pattern-equivalent, because $P^f(\bar{E})$ contains pattern $(0, 0, 2, 0, 1, 0)$ but $P^f(E)$ does not.

Proof. Let $E = (n, L, l)$ be an 1CSP instance with $P = P^b(E)$. Due to definition we have $0 < l_1 \le \cdots \le l_n \le L$. By multiplying the l_i and L with a suitable positive factor, we can ensure $l_1 \ge 1$. Similarly, we have $\sum_{i=1}^n l_i a_i \le L$ for all $a \in P$ and $\sum_{i=1}^n l_i a_i > L$ for all $a \in \mathbb{B}^n \setminus P$ by definition, so that multiplying the variables with a suitable positive factor ensures the validy of all constraints.

If L, l satisfy the inequalities (1), then $E = (n, L, l)$ is an example of the demanded instance. $\square$

We remark that we can additionally require that L and the l_i are positive integers and indeed we will use only integers in our subsequent examples of 1CSP instances.

The parameters l_i and L of Lemma 1 have the following nice geometric interpretation. The hyperplane defined by $\sum_{i=1}^n l_i x_i = L$ perfectly separates the set of feasible patterns $P^f(E) = P^p(E)$ and the set of non-feasible patterns $\mathbb{B}^n \setminus P^p(E)$ within the unit-hypercube. In Figure 1 we have depicted all five equivalence classes for $n = 3$, where the feasible patterns are marked by filled black circles.

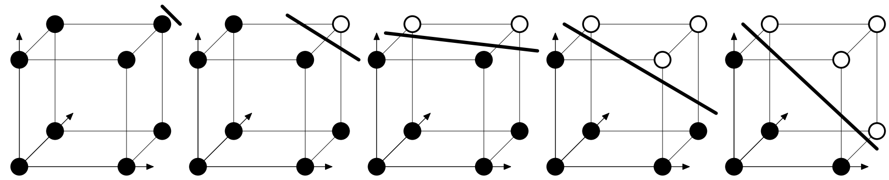

Figure 1: All equivalence classes of $\mathbb{P}_3^p$

So a first, simple but finite, algorithm to determine the maximum $\Delta_p(E)$ for given demand n, is to loop over all equivalence classes in $\mathbb{P}_n^p$ and to compute the respective $\Delta_p(E)$. Of course this is possible for rather small n only. Since it is of different interest to explicitly construct a complete system of representatives of $\mathbb{P}_n^p$, we present an enumeration algorithm in Section 5 before we proceed with ILP approaches in Section 7. Prior to that we relate our discrete structures with another stream of literature in the context of cooperative game theory.

4 Relation of 1CSP instances to weighted simple games

In cooperative game theory a *simple game* on n voters is defined as a mapping $v : \mathbb{B}^n \to \mathbb{B}$ satisfying $v(\mathbf{0}) = 0$, $v(\mathbf{1}) = 1$, and $v(a) \le v(b)$ for all $a, b \in \mathbb{B}^n$ with $a \le b$, i.e., $a_i \le b_i$ for all $1 \le i \le n$ (cf. [25]). A vector[2] $a \in \mathbb{B}^n$ with $v(a) = 1$ is called *winning* and *losing* otherwise. Each simple game is uniquely characterized by either its set of winning or its set of losing vectors. A simple game v is called *weighted* if there exist weights $w \in \mathbb{R}_{\ge 0}^n$ and a quota $q \in \mathbb{R}_{>0}$ such that $v(a) = 1$ iff $a^\top w \ge q$. W.l.o.g. we can assume that the quota and the weights are positive integers with $1 \le w_1 \le \cdots \le w_n \le q$ and $q \ge 2$.

[2]Mostly one speaks of subsets $S \subseteq \{1, \ldots, n\}$, called coalitions, in the corresponding literature. The vectors we use here correspond to the incidence vectors of those sets.

Given a weighted simple game v represented by weights $w \in \mathbb{Z}_{>0}$ and a quota $q \in \mathbb{Z}_{\geq 2}$, we can set $L = q - 1$ and $l_i = w_i$ for all $1 \leq i \leq n$. If additionally all unit-vectors are losing in v, then we have $l_i = w_i \leq q-1 = L$, i.e., $E = (n, L, l)$ is an 1CSP instance, where the losing vectors correspond to the feasible patterns.

For the other direction let $E = (n, L, l)$ be an 1CSP instance with $l \in \mathbb{Z}_{>0}^n$ and $L \in \mathbb{Z}_{>0}$. If additionally the all-one vector $\mathbf{1}$ is a non-feasible pattern, then setting $q = L + 1$ and $w_i = l_i$ for all $1 \leq i \leq n$ yields a weighted simple game v.

5 Enumeration of all pattern-equivalent classes of 1CSP instance

An equivalence class of an 1CSP instance E is uniquely described by its set $P = P^p(E) \subseteq \mathbb{B}^n$ of feasible patterns. So we have to enumerate all possible choices for P and subsequently decide which pattern is feasible and which is not. We observe that infeasibility in Lemma 1 may happen using only a proper subset of the inequalities (1).

Lemma 2. *Given two disjoint subsets $P_{\leq}$ and $P_{>}$ of $\mathbb{B}^n$. If*

$$\begin{aligned}
1 \leq l_1 \leq \cdots \leq l_n \leq L, \qquad & \qquad (2)\\
\sum_{i=1}^{n} l_i a_i \leq L \qquad & \forall a \in P_{\leq},\\
\sum_{i=1}^{n} l_i a_i \geq L + 1 \qquad & \forall a \in P_{>},\\
l_1, l_2, \ldots l_n, L \in \mathbb{R}_{\geq 0}. &
\end{aligned}$$

does not have a solution, then there can not exist an 1CSP instance $E = (n, L, l)$ with $P_{\leq} \subseteq P^p(E) \subseteq \mathbb{B}^n \setminus P_{>}$.

Next we observe, that some inequalities of (1) and (2) may be dominated by others. For $a \leq b$, i.e, $a_i \leq b_i$ for all $1 \leq i \leq n$, we clearly have $l^\top a \leq l^\top b$ due to $l \geq 0$. Using the special ordering $l_1 \leq \cdots \leq l_n$, we can even uncover more dominated inequalities. To this end we introduce the following binary relation.

Definition 2. *For $a, b \in \mathbb{B}^n$ we write $a \preceq b$ iff $\sum\limits_{i=j}^{n} a_i \leq \sum\limits_{i=j}^{n} b_i$ for all $1 \leq j \leq n$.*

We say that a is *dominated* by b. In the context of simple games the relation $\preceq$, using the reverse ordering of coordinates, is used to define the class of so-called complete simple games, which is a subclass of weighted simple games, see [9, 25]. So the following results are well known in a different context and we mention only the facts that we are explicitly using in this paper.

Lemma 3. *Let $a, b \in \mathbb{B}^n$ with $a \preceq b$. For $l_1 \leq \cdots \leq l_n$ we have $l^\top a \leq l^\top b$.*

Proof. Setting $l_j = \sum\limits_{i=1}^{j} k_i$, the $k_i \geq 0$ are uniquely defined and we have

$$l^\top a = \sum_{j=1}^{n} \left(k_j \cdot \sum_{i=j}^{n} a_i \right) \leq \sum_{j=1}^{n} \left(k_j \cdot \sum_{i=j}^{n} b_i \right) = l^\top b.$$

$\square$

Corollary 1. *Let* $\mathbf{0} \le a \preceq b \le \mathbf{1}$*. If* $b \in P^p(E)$*, then* $a \in P^p(E)$*. If* $a \notin P^p(E)$*, then* $b \notin P^p(E)$*.*

Lemma 4. $\mathbb{B}^n$ *is a partially ordered set under* $\preceq$*.*

Observation 1. $\{a \in \mathbb{B}^n : \|a\|_1 \le 1\} \subseteq P^p(E)$ *for all 1CSP instances* $E = (n, L, l)$ *due to* $l_i \le L$*.*

With those ingredients we can state the following enumeration algorithm.

`MainProcedure`()
$P_{\le} \leftarrow \{a \in \mathbb{B}^n : \|a\|_1 = 1\}$
$P_{>} \leftarrow \emptyset$
$P_u \leftarrow \mathbb{B}^n \setminus \{a \in \mathbb{B}^n : \|a\|_1 \le 1\}$
RecursiveProcedure($P_{\le},\ P_{>},\ P_u$)

`RecursiveProcedure`($P_{\le}$,$P_{>}$,P_u)
1 if system (2) has no solution for $P_{\le}$ and $P_{>}$
2 return
3 if $P_u = \emptyset$ $\triangleright$ we have found a new equivalence class
4 save $\{a \in \mathbb{B}^n : \exists b \in P_{\le} : a \preceq b\}$
5 return
6 choose some pattern $a \in P_u$ $\triangleright$ no matter which one
7 $P'_{\le} \leftarrow P_{\le}$
8 remove all patterns from $P'_{\le}$ which are dominated by a
9 $P'_u \leftarrow P_u \setminus \{b \in \mathbb{B}^n : b \preceq a\}$
10 RecursiveProcedure($P'_{\le} \cup \{a\},\ P_{>},\ P'_u$)
11 $P'_{>} \leftarrow P_{>}$
12 remove all patterns from $P'_{>}$ which dominate a
13 $P'_u \leftarrow P_u \setminus \{b \in \mathbb{B}^n : a \preceq b\}$
14 RecursiveProcedure($P_{\le},\ P'_{>} \cup \{a\},\ P'_u$)

Here $P_{\le}$ denotes a subset of the patterns that the algorithm has already classified as being feasible. Similarly, $P_{>}$ denotes a subset of the patterns that the algorithm has already classified as being non-feasible. Since patterns with a single element have to be feasible by definition, we can initialize as done in `MainProcedure`.

We know that a pattern a is feasible, if there exists a pattern $b \in P_{\le}$ with $a \preceq b$ – note that b is feasible. Similarly, we know that a pattern a is non-feasible, if there exists a pattern $b \in P_{>}$ with $b \preceq a$ – note that b is non-feasible. We remark that all unclassified patterns are pooled in P_u.

In order to save computation time within the check of Inequality system (2), we try to remove as many patterns as possible from $P_{\le}$ and $P_{>}$ in lines 8 and 12 of `RecursiveProcedure`. Since $\mathbb{B}^n$ is a partially ordered set under $\preceq$, the constructed sets $P_{\le}$ and $P_{>}$ are indeed minimal in every iteration. For $n = 9$ this approach reduces the computation time, due to a decreased number of inequalities in (2), by a factor of roughly 50.

The dominance relation $\preceq$ can clearly be checked in $O(n)$. Since those comparisons occur quite often it is beneficial to compute and store them once for all pairs of patterns. Some comparisons can additionally be avoided by using:

Observation 2. *For* $\mathbf{0} \le a \preceq b \le \mathbf{1}$ *we have* $\operatorname{num}(a) \le \operatorname{num}(b)$*, where* $\operatorname{num}(a) := \sum_{i=1}^{n} = 2^{i-1}a_i$.

The converse is generally not true, i.e., $\operatorname{num}(a) \le \operatorname{num}(b)$ implies either $a \preceq b$ or a and b are incomparable. In Figure 2 we have depicted the dominance relation, where patterns are ordered by the num() function. Black squares represent the cases $a \preceq b$; white ones the cases $a \not\preceq b$.

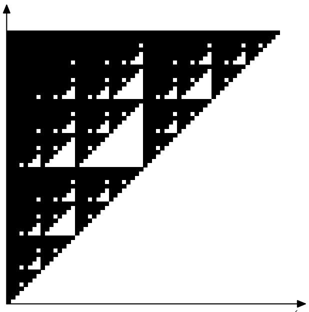

Figure 2: Illustration of the dominance relation

6 Bounds for $\Delta_p(E)$

Obviously we have $0 \le z_C^p(E) \le z_D^f(E) \le n$ for each 1CSP instance $E = (n, L, l)$. The cases with $z_D^f(E) = 1$ can be completely classified:

Lemma 5. *For an 1CSP instance* $E = (n, L, l)$ *we have*

$$z_D^f(E) = 1 \quad \Longleftrightarrow \quad \mathbf{1} \in P^p(E) \quad \Longleftrightarrow \quad P^p(E) = \mathbb{B}^n \quad \Longleftrightarrow \quad \sum_{i=1}^{n} l_i \le L.$$

Corollary 2. *If* $z_D^f(E) = 1$*, then we have* $z_C^p(E) = 1$ *and* $\Delta_p(E) = 0$.

Also the cases where $z_D^f(E) = 2$ can be characterized completely:

Lemma 6. *We have* $z_D^p(E) > 2$ *if and only if* $\{a, \mathbf{1}-a\} \not\subseteq P^p(E)$ *for all* $a \in \mathbb{B}^n$.

Proof. Choosing $a = \mathbf{1}$ we conclude $\mathbf{1} \notin P^p(E)$, since $\mathbf{0} \in P^p(E)$. Thus we can assume $z_D^f(E) > 1$. We have $z_D^f(E) = 2$ iff there exist feasible patterns $a, b \in P^p(E)$ with $a + b \ge \mathbf{1}$. Thus $b = \mathbf{1} - a \in P^p(E)$. $\square$

We remark that simple games, where not both coalition vectors a and $\mathbf{1} - a$ can be losing, are called *proper*.

The optimal solution of the proper relaxation is given by non-negative reel multipliers γ_a satisfying

$$\sum_{a \in P^p(E)} \gamma_a \cdot a = \mathbf{1} \quad \text{and} \quad z_C^p(E) = \sum_{a \in P^p(E)} \gamma_a. \tag{3}$$

Lemma 7. *$z_C^p(E) \geq 1$ for any instance E of 1CSP.*

Proof. From (3) we conclude

$$n \cdot \sum_{a \in P^b(E)} \gamma_a = \sum_{a \in P^p(E)} \gamma_a \cdot n \geq \sum_{a \in P^p(E)} \gamma_a \cdot \|a\|_1 = \|\mathbf{1}\|_1 = n.$$

□

The above proof can be slightly tightened if $\mathbf{1} \notin P^p(E)$.

Lemma 8. *If $z_D^f(E) = 2$, then $z_C^p(E) \geq \frac{n}{n-1}$ and $\Delta_p(E) \leq \frac{n-2}{n-1}$.*

Proof. Since $z_D^f(E) \neq 1$ we have $\mathbf{1} \notin P^p(E)$ so that $\|a\|_1 \leq n-1$ for all patterns $a \in P^p(E)$. Combining this with (3) yields

$$(n-1) \cdot \sum_{a \in P^p(E)} \gamma_a = \sum_{a \in P^p(E)} \gamma_a \cdot (n-1) \geq \sum_{a \in P^p(E)} \gamma_a \cdot \|a\|_1 = \|\mathbf{1}\|_1 = n.$$

Thus we have $z_C^p(E) \geq \frac{n}{n-1}$ and $\Delta_p(E) = z_D^f(E) - z_C^p(E) \leq \frac{n-2}{n-1}$. □

Lemma 9. *If $z_D^f(E) > 2$, then $z_C^p(E) > 2$ and $\Delta_p(E) < z_D^f(E) - 2$.*

Proof. W.l.o.g. we assume that the parameters L and l_i of $E = (n, L, l)$ are integers and that there exists a feasible pattern $a \in P^p(E)$ with $l^\top a = L$, since we may otherwise decrease L to obtain an equivalent representation with smaller L. From $z_D^f(E) > 2$ we conclude $\mathbf{1} - a \notin P^p(E)$ so that $l^\top(\mathbf{1} - a) > L$. Thus $L < \frac{1}{2} l^\top \mathbf{1}$. Multiplying equation (3) with vector l gives

$$\frac{l^\top \mathbf{1}}{2} \sum_{a \in P^p(E)} \gamma_a > \sum_{a \in P^p(E)} \gamma_a \cdot a^\top l = \mathbf{1}^\top l,$$

so that $z_C^p(E) = \sum_{a \in P^p(E)} \gamma_a > 2$. □

Corollary 3. *For $z_D^f(E) = 3$ we have $\Delta_p(E) < 1$.*

For $z_D^f(E) = 4$ we can also conclude $z_C^p(E) > 2$ and $\Delta_p(E) < 2$ from $z_D^f(E) \leq \frac{4}{3} \cdot \lceil z_C^p(E) \rceil$, see [3] for the later relative bound.

We remark that the instances with $z_D^p(E) \in \{n-1, n\}$ can be easily characterized. Since their number is in $O(n)$, we abstain from stating the details and provide exemplary enumeration results for $n = 8$ in Table 3.

Table 3: Number of equivalence classes for $n = 8$ and a given $z_D^p(E)$-value

$z_D^p(E)$	1	2	3	4	5	6	7	8
#	1	1363847	1277944	56895	1992	103	8	1

7 Integer linear programming approaches

Assume that we are not interested in all equivalence classes of 1CSP but only in those with $\Delta_p(E) \geq \delta$ for some parameter $\delta \geq 0$. For the search for proper non-IRUP instances we may set $\delta = 1$ and for the search of the largest possible $\Delta_p(E)$ for a given demand n we may update δ during a search algorithm. In the following subsections we present two algorithmic approaches.

7.1 A tailored branch-and-bound algorithm

We can easily convert the enumeration algorithm from Section 5 into a branch-and-bound algorithm with some additional cuts. To this end we state:

Lemma 10. *Let $E = (n, L, l)$ be an 1CSP instance and U, V be two subsets of $\mathbb{B}^n$ with $V \subseteq P^p(E) \subseteq U$. With this we have $z_C(U, E) \le Z_C(V, E)$ and $z_D(U, E) \le Z_D(V, E)$.*

Proof. Each feasible solution for pattern set V, i.e., each vector x with $A(V)x = \mathbf{1}$, can be extended to a feasible solution for pattern set U by inserting zeros for all patterns in $U \backslash V$. □

Corollary 4. *If $V \subseteq P^p(E) \subseteq U \subseteq \mathbb{B}^n$ for an 1CSP instance $E = (n, L, l)$, then $\Delta_p(E) \le z_D(U, E) - z_C(V, E)$.*

Our first modification of the enumeration algorithm is the extension of the lines 1 and 2 in `RecursiveProcedure` with the check from Corollary 4.

Depending on the chosen value of δ we can also utilize some of the bounds from Section 6 to start the algorithm with a non-empty set $P_>$. For, e.g., $\delta \ge 1$ we know $z_D^f(E) > 3$ so that we can set $P_> = \{a \in \mathbb{B}^n : \|a\|_1 = n - 2\}$ and remove each pattern $a \in \mathbb{B}^n$ with $\|a\|_1 \ge n - 2$ from P_u. Even more, every insertion of a pattern a into $P_\le$ may force some patterns to be non-feasible. If we can assume $z_D^f(E) > 2$, we especially have that $1 - a$ is non-feasible and can be put into $P_>$. Moreover, all patterns $b \succeq 1 - a$ are non-feasible and can be removed from P_u. As remarked, $\delta \ge 1$ implies $z_D^f(E) > 3$, so that $1 - c$ has to be non-feasible whenever there are feasible patterns a, b with $a + b = c$. Again, all patterns $a' \succeq 1 - c$ are non-feasible too and can be removed from P_u.

Because of the huge number of potential equivalence classes, the strategy of choosing pattern a from P_u in line 6 of `RecursiveProcedure` is really important. The best branching strategy we found is to choose a pattern $a \in P_u$ with the maximum positive multiplier in the optimal solution for the set of feasible patterns $\mathbb{B}^n \setminus \{a \in \mathbb{B}^n \ \exists b \in P_> : b \preceq a\}$. Sometimes the optimal solution has no intersection with P_u. In this case we can choose the branching pattern at random. Indeed this happens in less than 0.01% of all cases. This strategy reduces the search space of about 1000 times in comparison to a random choice.

The B&B algorithm presented above was implemented in C++, where we used a self implemented LP-solver with exact arithmetic. Making use of Intels Streaming SIMD Extensions and special shortcuts for our LP instances, our implementation of an LP-solver is about 30 times faster than the COIN-OR LP-solver.[3] As hardware we have used an Intel Core i7 with 4 GB RAM.

7.2 A direct integer linear programming formulation

Instead of implementing a tailored B&B algorithm one can also formulate the problem of the maximization of $\Delta_p(E)$ for a given demand n as an integer programming problem and use off-the-shelf ILP solvers. To this end we describe the set $P^p(E)$ of feasible patterns by binary variables $y_a \in \mathbb{B}$ for all $a \in \mathbb{B}^n$ and identify $P^p(E) = \{a \in \mathbb{B}^n : y_a = 1\}$. Partial information about $P^p(E)$ and

[3] Cf. [11], where the author also uses a self implemented LP solver to enumerate the weighted simple games with $n = 9$ voters.

$\mathbb{B}^n \setminus P^p(E)$ can be encoded by setting the variables of the respective patterns to either 1 or 0, respectively. Using the definition of an 1CSP instance only, we require $y_{\mathbf{0}} = 1$ and $y_{e_i} = 1$ for all $1 \le i \le n$.

To ensure the existence of the parameters L and l_i we have to further restrict the y_a. Given an upper bound M on L, the inequalities of Lemma 1 can be formulated using so-called Big-M constraints. Fortunately all this is already known in the context of weighted simple games, see [12, 14]. So, without any further justification we state that the 1CSP instances with demand n are in one-to-one correspondence to the feasible 0/1 solutions y of:

$$
\begin{aligned}
y_{\mathbf{0}} &= 1 && \\
y_{e_i} &= 1 && \forall\, 1 \le i \le n \\
y_a - y_b &\ge 0 && \forall\, a, b \in \mathbb{B}^n : \, a \preceq b \\
\sum_{i : a_i = 1} l_i &\le L + (1 - y_a) \cdot M && \forall a \in B^n \\
\sum_{i : a_i = 1} l_i &\ge L + 1 - y_a \cdot M && \forall a \in B^n \\
l_i &\le l_{i+1} && \forall\, 1 \le i < n \\
l_n &\le L && \\
y_a &\in \mathbb{B} && \forall a \in \mathbb{B}^n \\
L, l_i &\in \mathbb{Z}_{\ge 1} && \forall\, 1 \le i \le n,
\end{aligned}
$$

where M can be chosen as $4n\left(\frac{n+1}{4}\right)^{(n+1)/2}$.

In principle we would like to maximize the target function $z_D^f(E) - z_C^p(E)$. Unfortunately both terms are the optimal values of optimization problems itself. Since the later term arises from an LP we can model optimality by using the duality theorem, see [7] for an application of this technique in the context of simple games. Here it is even simpler since we can even take any feasible solution of the LP of $z_C^p(E)$ due to the maximization. So we replace $z_C^p(E)$ by $\sum_{a \in \mathbb{B}^n} x_a$ and add the constraints

$$
\begin{aligned}
\sum_{a \in \mathbb{B}^n : \, a_i = 1} x_a &= 1 && \forall\, 1 \le i \le n \\
x_a &\le y_a && \forall\, a \in \mathbb{B}^n \\
x_a &\in \mathbb{R}_{\ge 0} && \forall\, a \in \mathbb{B}^n
\end{aligned}
$$

For $z_D^f(E)$ this approach does not work, since there is no duality theorem for ILPs and $z_D^f(E)$ has a different sign as $z_C^p(E)$ in the target function. So we choose a different approach. By introducing further inequalities we can ensure that $z_D^f(E) \ge k$ holds for all feasible solutions, where k is an arbitrary but fixed integer. Let y_a^i be additional binary variables, which equal 1 if pattern $a \in \mathbb{B}^n$ can be written as the sum of at most $1 \le i < k$ feasible patterns a^j, where $j = 1, \dots, i$. For $i = 1$ we have $y_a^1 = y_a$ for all $a \in \mathbb{B}^n$. Next we require $y_a^i \ge y_a^{i-1}$ for all $a \in \mathbb{B}^n$, $2 \le i < k$ and

$$
y_a^i \ge y_u^{i-1} + y_v^1 - 1 \quad \forall\, a, u, v \in \mathbb{B}^n : \, u + v = a \text{ and } \forall\, 2 \le i < k.
$$

As a justification let us consider a pattern $a \in \mathbb{B}^n$ that can be written as the sum of at most i feasible patterns. If there exists such a representation with at most $i-1$ summands, i.e. $y_a^{i-1}=1$, then $y_a^i \geq y_a^{i-1}$ implies $y_a^i=1$. Otherwise there exists a feasible pattern v and a pattern u, that can be written as the sum of at most $i-1$ feasible patterns, with $a=u+v$. Thus, $y_u^{i-1}=1$, $y_v^1=1$, and so also $y_a^i=1$. We remark that $y_a^i=1$ is also possible, if pattern a can not be written as the sum of at most i feasible patterns.

With these extra variables at hand, requiring $y_{(1,\dots,1)}^{k-1}=0$ guarantees $z_D^f(E) \geq k$. Note that a sum $\sum_{j=1}^{i} a^j \geq a$ of feasible patterns a^j implies the existence of feasible patterns $\tilde{a}^j$ with $\sum_{j=1}^{i} \tilde{a}^j = a$.

Many of these inequalities are redundant, which is found out quickly by a customary ILP solver. We remark that it is not necessary to consider variables y_a^i for all $i \in \{1,\dots,k-1\}$. By using inequalities of the form $y_a^i \geq y_u^{i_1} + y_v^{i_2} - 1$, where $i_1 + i_2 = i$, $i_1, i_2 < i$, $\Theta(\log k)$ values for i are sufficient in general. Nevertheless the ILP model becomes quite huge, so that we solved it with the Gurobi ILP solver on an Intel Xeon with 384 GB RAM. Of course one may try to deploy more sophisticated ILP techniques like column generation or cut separation, which goes beyond the scope of this paper.

8 Computational results

For $n \leq 9$ we have used the enumeration algorithm from Section 5 to generate all 1CSP instances with demand n. In Table 1 we have stated the number $|\mathbb{P}_n^p|$, the maximum value $\Delta_p(E)$, the number and the corresponding list of instances (representatives of equivalence classes)) attaining this maximum value, whenever computationally possible. For each mentioned instance we have used the smallest possible integer valued parameters l_i and L.

The stated results for $n=10,11$ are obtained with the B&B-algorithm of Subsection 7.1 setting δ to 1 or $1+\varepsilon^4$, respectively. The computation time for $n=11$ was 17 hours.

For the cases $12 \leq n \leq 14$ we restricted the search to classes with large values of $z_D^p(E)$ due to the exponential growth of $|\mathbb{P}_n^p|$. For $n=12$ we checked all equivalence classes with $z_D^p(E) \geq 5$, for $m=13$ only $z_D^p(E) \geq 6$, and for $m=14$ only $z_D^p(E) \geq 7$.

Using the ILP formulation of Subsection 7.2 (and suitable bounds from Section 6) we have verified the maximum Δ_p-value for $n \leq 11$, while consuming a considerably larger amount of computation time.

By slightly modifying the constraints of Lemma 2, according to the remarks in Section 4, we have also computed the number of weighted simple games with up to $n=9$ voters. This uncovers read-write disk failures within the computation done in [11], so that the number of weighted simple games for $n=9$ voters was corrected from 989913344 to 993061482.

9 Conclusion

We have presented an enumeration algorithm for all equivalence classes of 1CSP instances. For a demand of at most 9 the corresponding numbers are determined. As a side result we could correct the number of weighted simple games for 9

[4] Each $0 < \varepsilon \leq \frac{1}{125}$ would have worked.

Table 1: Results of computational experiments

m	$\lvert\mathbb{P}^b_m\rvert$	$\max \Delta_p$	*	instances from classes with maximum Δ_p
1	1	0	1	$L = 1, l = (1)$
2	2	0	2	$L = 1, l = (1, 1); L = 2, l = (1, 1)$
3	5	1/2	1	$L = 2, l = (1, 1, 1)$
4	17	2/3	1	$L = 3, l = (1, 1, 1, 1)$
5	92	3/4	2	$L = 4, l = (1, 1, 1, 1, 1); L = 4, l = (1, 1, 2, 2, 3)$
6	994	7/8	1	$L = 8, l = (1, 2, 2, 3, 4, 5)$
7	28262	16/17	1	$L = 17, l = (2, 3, 4, 5, 6, 7, 8)$
8	2700791	38/39	1	$L = 39, l = (2, 5, 6, 8, 11, 14, 15, 18)$
9	990331318	103/104	2	$L = 104, l = (7, 12, 16, 19, 22, 27, 30, 36, 40)$ $L = 104, l = (11, 15, 18, 20, 24, 27, 28, 32, 34)$
10		1	365	$L = 81, l = (4, 6, 6, 9, 16, 29, 32, 37, 40, 62)$ $L = 89, l = (4, 6, 7, 10, 18, 32, 35, 41, 44, 68)$ $L = 101, l = (5, 7, 8, 11, 20, 36, 40, 46, 50, 78)$ $L = 142, l = (7, 10, 11, 16, 28, 51, 56, 65, 70, 108)$ and 361 other instances
11		126/125	6	$L = 155, l = (9, 12, 12, 16, 16, 46, 46, 54, 69, 77, 102)$ $L = 193, l = (11, 15, 15, 20, 20, 57, 58, 67, 86, 96, 127)$ $L = 204, l = (12, 16, 16, 21, 21, 60, 61, 71, 91, 101, 134)$ $L = 207, l = (12, 16, 16, 21, 22, 61, 62, 72, 92, 103, 136)$ $L = 218, l = (13, 17, 17, 22, 23, 64, 65, 76, 97, 108, 143)$ $L = 221, l = (13, 17, 17, 23, 23, 65, 66, 77, 98, 110, 145)$
12		31/30 **		$L = 18, l = (4, 4, 6, 6, 6, 7, 7, 9, 9, 9, 10, 12)$
13		53/50 **		$L = 34, l = (8, 8, 10, 11, 11, 12, 12, 13, 13, 17, 17, 17, 18)$ $L = 48, l = (11, 11, 14, 15, 16, 17, 17, 18, 19, 24, 24, 24, 25)$
14		17/16 **		$L = 42, l = (7, 7, 10, 10, 12, 15, 15, 21, 21, 21, 22, 22, 28, 31)$ $L = 50, l = (8, 9, 12, 12, 14, 18, 18, 25, 25, 25, 26, 26, 33, 37)$

* — number of classes with maximum Δ_p
** — maximum found gap, without computational proof of optimality

voters (incorrectly) stated in [11]. To the best of our knowledge, the relation between 1CSP instances and weighted simple games is indicated for the first time. By enhancing the enumeration approach to a B&B algorithm we were able to computationally prove that all 1CSP instances with demand of at most 9 are proper IRUP instances, while we found classes of non-IRUP instances with demand $n = 10$ and $\Delta_p = 1$. This resolves an open question from [2, 4], where the authors ask for proper non-IRUP instances with $n < 13$. $\Delta_p > 1$ is possible for $n \geq 11$ only. Even more we have exactly determined the maximum proper gap Δ_p for $n \leq 11$ and classified all instances attaining the maximum gap. For further investigations on the structure of 1CSP instances with large gap, we have made them available at http://www.math.tu-dresden.de/∼capad/capad.html. By partially going through the search space for $n \geq 12$, we improved the worst known proper gap from 1.003 to 1.0625.

With respect to the exact value $z^p_D(E)$ we have proven that all 1CSP instances with $z^p_D(E) \leq 3$ are proper IRUP instances with a proper gap smaller than 1, while there are examples with $z^p_D(E) = 4$ having $\Delta_p(E) > 1$.

Focusing on the size of L, and so indirectly on the size of the l_i, we mention that the first known constructions of proper non-IRUP instances were rather huge. The example of [16] has $L = 3, 397, 386, 355$ and was decreased to just $L = 1, 111, 139$ in [3]. Recently the authors of [2, 4] gave an example with $L = 100$. Our smallest found example has $L = 18$. It would be nice to know whether this is best possible.

We leave the famous (proper) MIRUP conjecture still widely open and encourage more research in that direction.

With respect to the enumeration of weighted simple games, the case of $n = 10$ voters might be in range of the presented exhaustive algorithm if further tuned. Some adaptation towards the inverse power index problem, see [12, 13, 14], is imaginable too.

Acknowledgements

The auhors would like to thank two anonymous referees for very valuable remarks on an earlier draft. The research was supported by the Russian Foundation of Basic Research (RFBR), project 12-07-00631-a.

References

[1] S. Baum and L.E. Trotter, Jr. Integer rounding for polymatroid and branching optimization problems. *SIAM Journal on Algebraic Discrete Methods*, 2(4):416–425, 1981.

[2] A. Caprara, J.C. Díaz Díaz, M. Dell'Amico, M. Iori, and R. Rizzi. Friendly bin packing instances without integer round-up property. Technical report, Università di Modena e Reggio Emilia, 2012. submitted.

[3] L.M.A. Chan, D. Simchi-Levi, and J. Bramel. Worst-case analyses, linear programming and the bin-packing problem. *Mathematical Programming*, 83(1-3):213–227, 1998.

[4] J.C Díaz Díaz. *Exact and heuristic solutions for combinatorial optimization problems*. PhD thesis, Università Degli Studi di Modena e Reggio Emilia, 2012.

[5] F. Eisenbrand, D. Pálvölgyi, and T. Rothvoß. Bin packing via discrepancy of permutations. *ACM Transactions on Algorithms (TALG)*, 9(3):24, 2013.

[6] M. Fieldhouse. The duality gap in trim problems. *SICUP Bulletin*, 5(4), 1990.

[7] J. Freixas and S. Kurz. On α-roughly weighted games. *International Journal of Game Theory*, pages 1–34, 2014. To appear, doi:10.1007/s00182-011-0308-4.

[8] P.C. Gilmore and R.E. Gomory. A linear programming approach to the cutting-stock problem. *Operations research*, 9(6):849–859, 1961.

[9] J.R. Isbell. A class of majority games. *The Quarterly journal of mathematics*, 7(1):183–187, 1956.

[10] V.M. Kartak. Sufficient conditions for the integer round-up property to be violated for the linear cutting stock problem. *Automation and Remote Control*, 65(3):407–412, 2004.

[11] S. Kurz. On minimum sum representations for weighted voting games. *Annals of Operations Research*, 196(1):361–369, 2012.

[12] S. Kurz. On the inverse power index problem. *Optimization*, 61(8):989–1011, 2012.

[13] S. Kurz. The inverse problem for power distributions in committees. Technical report, University of Bayreuth, 2013. submitted, 46 pages, available at http://arxiv.org/abs/1402.0988.

[14] S. Kurz and S. Napel. Heuristic and exact solutions to the inverse power index problem for small voting bodies. *Annals of Operations Research*, 215(1):137–163, 2014.

[15] O. Marcotte. The cutting stock problem and integer rounding. *Mathematical Programming*, 33(1):82–92, 1985.

[16] O. Marcotte. An instance of the cutting stock problem for which the rounding property does not hold. *Operations Research Letters*, 4(5):239–243, 1986.

[17] C. Nitsche, G. Scheithauer, and J. Terno. Tighter relaxations for the cutting stock problem. *European Journal of Operational Research*, 112(3):654–663, 1999.

[18] J. Rietz and S. Dempe. Large gaps in one-dimensional cutting stock problems. *Discrete Applied Mathematics*, 156(10):1929–1935, 2008.

[19] J. Rietz, G. Scheithauer, and J. Terno. Families of non-irup instances of the one-dimensional cutting stock problem. *Discrete Applied Mathematics*, 121(1):229–245, 2002.

[20] J. Rietz, G. Scheithauer, and J. Terno. Tighter bounds for the gap and non-irup constructions in the one-dimensional cutting stock problem. *Optimization*, 51(6):927–963, 2002.

[21] G. Scheithauer and J. Terno. About the gap between the optimal values of the integer and continuous relaxation one-dimensional cutting stock problem. In *Operations Research Proceedings 1991*, pages 439–444. Springer, 1992.

[22] G. Scheithauer and J. Terno. A branch-and-bound algorithm for solving one-dimensional cutting stock problems exactly. *Applicationes Mathematicae*, 23(2):151–167, 1995.

[23] G. Scheithauer and J. Terno. The modified integer round-up property of the one-dimensional cutting stock problem. *European Journal of Operational Research*, 84(3):562–571, 1995.

[24] G. Scheithauer and J. Terno. Theoretical investigations on the modified integer round-up property for the one-dimensional cutting stock problem. *Operations Research Letters*, 20(2):93–100, 1997.

[25] A.D. Taylor and W.S. Zwicker. *Simple games: desirability relations, trading, pseudoweightings.* Princeton University Press, 1999.